Appl. No. 10/707,645
Amdt. dated June 29, 2006
Reply to Office action of April 14, 2006

Amendments to the Specification:

Please replace paragraph 0007 with the following amended paragraph:

When the control circuit 28 is performing the ~~aaddress~~ address assignment, the apparatus 16 will view all the memory units in the memory modules as a ~~wwhole~~ whole. Therefore, when the control circuit 28 is assigning addresses to units in the memory module 30B, the value of the address will continuously be increased ~~incrementallyfrom~~ incrementally from the address 36E, which is the largest address in the memory module 30A. As shown in Fig. 2, the first memory unit of the memory module 30B will be assigned the address 38A; the value of the address 36E will be increased by one to "0...010...0" (only Bit 25 is '1'), which represents the $(2^{25}+1)^{th}$ memory unit of the memory apparatus 16 -- that is, the $(2^{25}+1)^{th}$ memory unit counting from the first memory unit of the memory module 30A (the memory unit associated with the address 36A). Similarly, the second memory unit in the memory module 30B will be viewed as the $(2^{25}+2)^{th}$ memory unit in the memory apparatus 16, and its associated address 38B will be increased by one from address 38A to a binary number of "0...010...01." (Only Bit 0 and Bit 25 are '1'.) Since there are 2^27 memory units in the memory module 30B, the last two memory units in the memory module 30B will become the $(2^{25}+2^{27}-1)^{th}$ and the $(2^{25}+2^{27})^{th}$ memory units in the memory apparatus 16, and their associated addresses 38C and 38D will be increased respectively to "0...01001...10" (Bit 1 to Bit 24 and Bit 27 are '1', and the rest are '0') and "0...01001...11" (Bit 0 to Bit 24 and Bit 27 are '1' and the rest are '0').

Please replace paragraph 0024 with the following amended paragraph:

The present invention decodes a given address in a memory device by comparing mutually exclusive bit-patterns of addresses. By using the sorting technique disclosed in the present invention, each memory module can be associated with one unique address based on its memory size while each address has its own mutually exclusive bit-pattern.

In other words, for all addresses located in the same memory module, the value of some particular ~~bbits isfixed~~ bits is fixed, such that forming a common address shared by every address in the memory module. The common addresses associated with any two different memory modules will be exclusive. (That is, there must be at least a one-bit difference between common addresses of any two different memory modules.) For comparing whether the given address matches any common address of those memory modules, the present invention actually compares whether the value of a particular bit in the given address matches a fixed predetermined one. Since the comparison of patterns are implemented in the way of comparing whether the value of each bit of the given address matches its associated bit in the common address, the comparison does not require ripple carrying between different bits in the process of performing the addition. Thus far, the technique in the present invention can perform initial address decoding rapidly and efficiently so as to increase the access speed and efficiency of memory resources and the computing power of the computer as a whole.

Please replace paragraph 0025 with the following amended paragraph:

Please refer to Fig. 4. Fig. 4 is a memory address allocation diagram showing how the address is assigned to each memory module. To facilitate the comparison with the prior art in Fig. 2, Fig. 4 makes an assumption that the memory module 80A to 80D in the present invention has a memory size of ~~34MB~~ 32MB, 128MB, 256MB and 64MB respectively, that is, 2^25, 2^27, 2^28 and 2^26 memory units. The present invention will assign the linearly increased 32-bit binary address to each memory unit. However, the present invention will assign the address based on the memory size of each memory module. The basic principle of the present invention is that the larger the memory size of a memory module is, the smaller its assigned binary address will be.

Please replace paragraph 0035 with the following amended paragraph:

Please refer to Fig. 6. Fig. 6 is a functional block diagram that takes the comparison unit 112B as example; the function of the comparison unit 112B is used for comparing whether the given address matches the bit-pattern 96B. As described in Fig. 4, the present invention can deduct the associated standard address and mask from each bit-pattern, and while implementing the comparison unit 112B, the mask 99B and the standard address 98B can be associated with the bit-pattern 96B. As shown in Fig. 6, the comparison unit 112B can be equipped with a plurality of AND gates and NXOR gates. Each AND gate is used for ANDing one bit in the given address 104 with another bit in the mask 99B. For example, AND gates 118A to 118G are used for ANDing Bit 31 to Bit 25 in the given address 104 with Bit 31 to Bit 25 in the mask 99B. The NXOR Gates are used for further NXORing the result of each AND operation with another bit in the standard address. For example, NXOR gates 120A to 120G in Fig. 6 are used for NXORing the result of each AND operation of the gates 118A to 118G and one bit in the standard address. The results output from each NXOR gate will go through the AND gate 122 again, and the comparison result of the comparative unit 112B will be out from the AND gate 122. While ANDing each bit in the mask with the given address 104, each AND gate "masks" those bits not required for comparison in the given address 104 and transmits the value of bits required for comparison to the NXOR gates; while NXORing the result from each AND gate with each bit in the standard address, the NXOR gates compares whether the bits required for comparison in the given address 104 is equal to the associated bits in the standard address; the AND gate 122 integrates the outputs from all of the NXOR gates. Take the example shown in Fig. 6, the value of each bit from Bit 27 to Bit 31 in the mask 99B is actually '1', which stands for that the value of Bit 27 to Bit 31 in the associated bit-pattern 96B is fixed. The comparison unit 112B will check whether Bit 27 to Bit 31 in the given address 104 matches Bit 27 to Bit 31 in the standard address 98B respectively. The value '1' from Bit 31 to Bit 27 in the mask 99B will result in the output of the AND gate 118A to 118E being determined by Bit 31 to Bit 27 in the given address 104. The

result is in effect equal to transmitting each value of Bit 31 to Bit 27 in the given address 104 to the NXOR gate 120A to 120E respectively. Furthermore, the NXOR operation performed by the NXOR gates 120A and 120E is in effect equal to comparing whether Bit 27 to Bit 31 in the given address 104 matches Bit 27 to Bit 31 in the standard address 98B respectively. If equal, the AND gate 122 will have an output of "true." In comparison, given that Bit 26 and Bit 25 in the mask 99B are '0', the outputs of the AND gate 118F and 118G will be "false" no matter what the values of Bit 26 and Bit 25 in the given address 104 are; this is in effect equal to masking Bit 26 and Bit 25 in the given address 104. The outputs of "false" from the AND gates 118F and 118G along with the values of '0' of Bit 26 and Bit 25 in the standard address 98B will make the outputs of the NXOR gate 118F and 118G to always be "true." This will result in the NXOR gates 120A to 120E controlling the output of the AND gate 122. Fig. 6 also makes the assumption that the value of the given address 104 is same as the value in Fig. 5; under the circumstance, since the value of Bit 27 in the given address 104 is '1', which does not match the value of '0' of Bit 27 in the standard address 98B, the output of the NXOR gate 120E is "false." This will result in the output of the AND gate 122 in the comparison module 112B being "false." ~~"ffalse."~~ In Fig. 5 and Fig. 6, each functional block in the present invention can be implemented either by hardware, software program in the micro-control circuit, or a mix of these two kinds. For example, the sorting module 116 and the logic module 100 in Fig. 5 can be implemented by software. Though the comparison units are implemented as AND gates and NXOR gates in Fig. 6, they can also be implemented in hardware. Therefore, while executing the functional block in Fig. 6 and Fig. 5, the software program can be stored in the BIOS 24. (Please refer to Fig. 1). From the discussion of embodiment above, the present invention performs initial address decoding by comparing bit-patterns so as to enable a fast and efficient embodiment. Because while each comparison unit is performing bit-pattern comparison, it is actually comparing whether the bits of fixed values in the bit-pattern actually match the associated bits in the given address 104. The comparison of those bits required for it is done in parallel and integrated swiftly. For

example, while the comparison unit 112B in Fig. 6 is operating, the present invention can check whether Bit 25 to Bit 31 in the given address 104 actually matches the associated bits in the standard address 98B and AND the result of comparison together. In terms of the operation time of digital circuits, since each comparison unit in the present invention can fulfill the mask operations for each bit in the given address 104 at the same time and the AND operation for the result of comparison, the amount of time required for the process above is approximately: the time required for the mask operation of a single bit in the AND gate, the time required for the comparison of a single bit in the NXOR gate, plus the time required for the integration of the result of each NXOR gate. In the three different operations above, each of them may be implemented with a single logic gate. Therefore, the present invention can accomplish the operation of the whole comparison module swiftly.

Please the abstract with the following amended paragraph:

A memory address decoding method for determining if a given address is located in one of a plurality of sections. Each section has a plurality of memory units and each memory unit has a unique corresponding address, the corresponding address using the binary system. The method includes making the corresponding address in a section with greater size smaller than the corresponding address in a section with smaller size, ~~bbuilding asingle~~ <u>building a single</u> bit-pattern for each section from all corresponding addresses, and comparing if at least one comparative bit of the given address matches those in any of the bit-patterns so as to determine the given address is located in one of the sections based on the comparison.